Video Resolution Enhancement with Multiple Motions

by

Raynard Orin Hinds

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1994 [Feb 1995]

Author .....
Department of Electrical Engineering and Computer Science
May 6, 1994

Certified by .....
Walter Bender
Principal Research Scientist, MIT Media Laboratory
Thesis Supervisor

Certified by .....
Alex Drukarev
Project Manager, Hewlett-Packard Company
Thesis Supervisor

Accepted by .....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Video Resolution Enhancement with Multiple Motions

by

Raynard Orin Hinds

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 1994, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Master of Science

Abstract

The goal of video resolution enhancement is to produce a single high resolution still image from several lower resolution images. With multiple image frames, sub-pixel motion in video sequence can be exploited to produce a single high-resolution frame. Motion estimation is done between frames and the information contained in each low resolution frame is combined.

This thesis develops a video resolution enhancement method for scenes containing multiple independently moving objects and evaluates the results. A comparison is made to several interpolation methods which use only a single image frame and establish a reference point for video resolution enhancement.

Thesis Supervisor: Walter Bender
Title: Principal Research Scientist, MIT Media Laboratory

Thesis Supervisor: Alex Drukarev
Title: Project Manager, Hewlett-Packard Company

The work reported herein was supported by Hewlett-Packard Company

Acknowledgments

I would like to thank my thesis advisors Alex Drukarev and Walter Bender for their guidance and support throughout this project.

I would also like to thank the members of the Imaging Technology Department at Hewlett-Packard Laboratories for their knowledgeable advice and feedback. The insights provided, enriched my experience and encouraged the completion of this project.

I would like to give special thanks to my parents and brother for their unconditional support of me both now and while growing up. In addition, I want to thank my aunt, uncle, and family for providing me with all the comforts and support of home while working on this project.

Contents

1 Introduction **6**

1.1 Problem Formulation 7

1.1.1 Interpolation 8

1.1.2 Multi-frame Resolution Enhancement 8

1.2 Thesis Overview 11

2 Interpolation **13**

2.1 Nearest-Neighbor Interpolation 14

2.1.1 Example 15

2.2 Bilinear interpolation 15

2.3 Cubic interpolation 16

2.3.1 Piecewise Cubic Convolution 21

2.3.2 Results 23

3 Multiple Motion Estimation **25**

3.1 Motion Estimation Overview 26

3.1.1 Block Matching 26

3.1.2 Region-Based Motion Estimation 27

3.1.3 Pixel-Based Motion Estimation 29

3.1.4 Robust Estimation 32

4 High-Resolution Reconstruction **44**

4.1 High-Resolution Reconstruction 44

4.2 Example of a Synthetic Image 47

4.2.1 Interpolation 48
4.2.2 Multi-Frame Resolution Enhancement 48
4.3 Examples of Real Images 50
4.3.1 Reduction of Image Blurring 51
4.3.2 Two Motions 51
4.3.3 Three Motions 52

5 Conclusion **64**

Chapter 1

Introduction

A video sequence consists of still image frames displayed in rapid succession. The resolution of a single frame is poor due to the limitations of the camera sensor. The rapid rendition of video scenes often masks the resolution loss, as well as other degradations from the imaging process. When a single frame is displayed, there is noticeable loss of information and detail from the original scene. As a result, it is difficult to produce a high quality still image using a video sequence frame. In the absence of the opportunity to modify the camera sensor, improving the resolution of a single frame can only use the information present in the video sequence. This thesis examines different approaches to video resolution enhancement.

Resolution enhancement of video frames is a valuable tool. Many events involve dynamic activity and motion which are best captured in video. At the same time, still photographs of events are often desired. It is very cumbersome to obtain both forms of image capture. Often a photographer is forced between video and still frame cameras to best preserve the memories of the event. Higher resolution, beyond the resolution of any single video frame, is needed to make scenes captured from video comparable to those photographed with still cameras. In addition, large format prints of the image may be desired and the size of the image must increase.

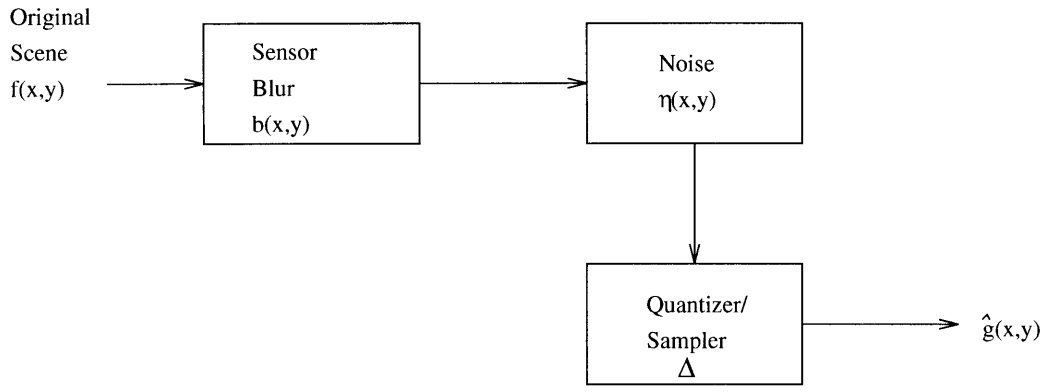

Figure 1-1: Imaging process

1.1 Problem Formulation

The goal of video resolution enhancement is to produce a single high-resolution still image from several lower-resolution video frames. To understand the problem of resolution enhancement, it is first important to understand how the original images are formed. The imaging process is shown in Figure 1-1. The original scene can be described as a two dimensional continuous function $f(x, y)$. The scene is first imaged through the camera sensors. Since the response of the sensor is bandlimited, this process blurs the image of $f(x, y)$. This is equivalent to convolving the input signal $f(x, y)$ with the blurring function of the camera $b(x, y)$. As in all real systems, the resulting image is corrupted by additive noise $\eta(x, y)$. To process and print the image using computers, the resulting image signal must be quantized and sampled (represented by $\Delta()$) resulting in the sampled digital signal $\hat{g}(x, y)$. In this case, x and y represent the discrete sampled locations of the continuous image $f(x, y)$.

The imaging process can be expressed mathematically as follows [1]:

$$\hat{g}(x,y) = \Delta(\sum_{(i,j)\epsilon R_b} b(i,j)f(x-i,y-j) + \eta(x,y)) \tag{1.1}$$

Once the signal is digitized, the picture elements (pixels) from a low-resolution video frame can be considered as samples of the scene photographed. There are 2 possible techniques to produce higher resolution still frames: single frame image interpolation and multi-frame resolution enhancement. Possible applications of this research include printing high resolution stills of an NTSC frame, large format printing, transform of NTSC video to HDTV quality video, and low bit rate video coding. This thesis does not address the problem of noise reduction.

1.1.1 Interpolation

Image interpolation is the simplest way to increase the number of sample points of an image. Interpolation attempts to derive the original continuous signal, which can be resampled at smaller intervals, using only information from the frame to be enhanced.

It does not truly increase the resolution, since no more information is added to the image. However, increasing the number of sample points and restoring the image by deblurring, will result in an image which appears better than the original.

1.1.2 Multi-frame Resolution Enhancement

Using a single frame to increase the resolution is difficult if not impossible. Removing the restriction of only one frame may facilitate an actual resolution increase. A true increase in resolution of an image requires 2 tasks:

1. More information must be obtained from the scene. The degraded image frame is lacking detail and information that can be resolved by the viewer from the

[1] $R_{variable}$ refers to the region of support of the *variable*.

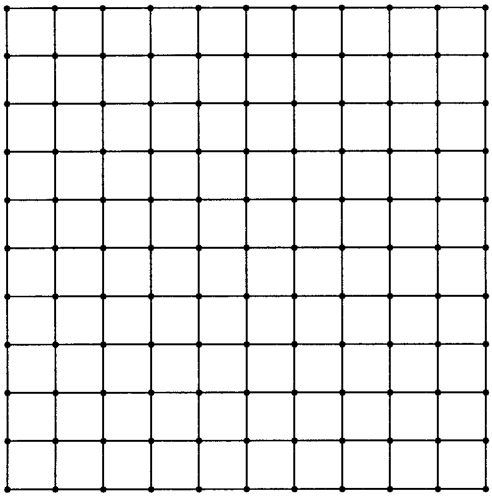

Figure 1-2: Sampling grid of digitized video frame

scene. Resolution enhancement attempts to restore this lost information.

2. The information must be correctly incorporated into the low-resolution image. It is not adequate to only increase the information from the scene, but this information must be combined with the degraded image to produce a higher-resolution image.

Video sequences tend to contain scenes with moving objects, as well as global camera motion. A single frame of a video sequence produces a static image which when digitized can be considered as coarse samples of the original scene (Figure 1-2). With the sub-pixel motion between frames, the samples obtained in successive frames are from slightly shifted scene positions relative to the original scene samples. Thus, additional frames from a video sequence increase the amount of information obtained from the scene.

To increase the resolution of a single frame, additional information from successive video frames can be added. To correctly incorporate additional information to the degraded frame, motion estimation techniques must be used between the original reference frame and successive video frames. In this manner, the precise sub-pixel displacement of samples from successive frames is determined in the reference frame.

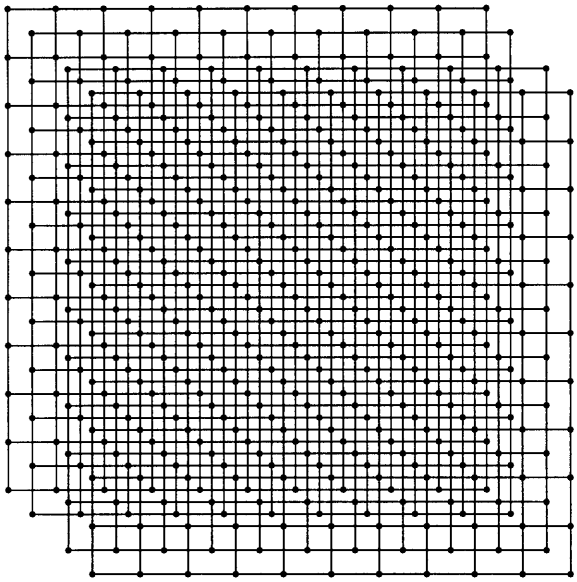

Figure 1-3: 4 registered frames with global motion.

Solutions to this problem have been attempted using images with global camera motion [10, 13, 14, 15, 18]. In this case, all the motion between two frames is the same for each point in the image, and it is easier to do accurate motion estimation and registration. Different processing techniques have been proposed to reconstruct the high-resolution frame. Once the global camera motion is estimated, the frames can simply be aligned on a high-resolution grid and the resolution of overlapping regions is increased (Figure 1-3).

In reality, video sequences are not restricted to only global camera motion. Most video scenes taken in natural environments consist of different people and objects moving in independent directions. This thesis attempts resolution enhancement on frames from video sequences containing multiple motions generated by several differently moving objects. The difficult task is to obtain accurate motion estimation. In Figure 1-4, the information in the second frame registers differently for each object. Once the sub-pixel displacements are known for each object, the information from moving objects can be aligned on a higher resolution grid.

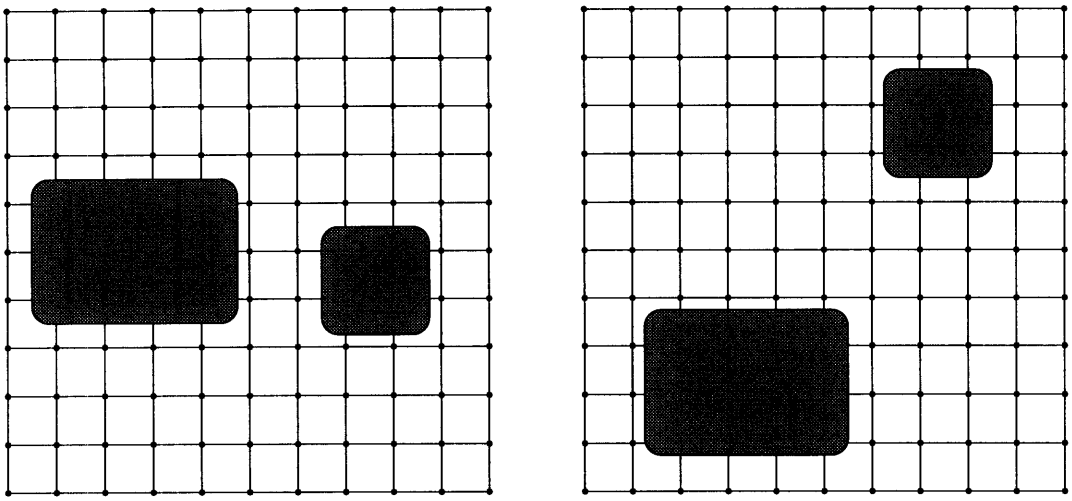

Figure 1-4: 2 frames with multiple motion.

1.2 Thesis Overview

Chapter 2 investigates several single frame interpolation methods. Experiments are done to determine the results of simple interpolation to establish a reference point for multi-frame resolution enhancement. Presumably, any resolution enhancement algorithm which uses multiple frames will be more complex. To make the effort worthwhile, the results should be better than the simple interpolation technique.

Chapter 3 examines several algorithms for motion estimation proposed in literature. It is important that the motion estimation algorithm be able to successfully estimate the motion in scenes containing several differently moving objects. At the same time, computational intensity must be monitored.

Once velocity estimates for the motion in a video sequence are obtained, the high-resolution image can be reconstructed. Interpolation and restoration are also needed in multi-frame resolution enhancement to produce the final image. Interpolation is needed, since the sample point values are collected on grids which have random relative shifts. This results in overall non-uniform sampling. Thus, some points will have to be interpolated onto the high-resolution grid to reconstruct the higher-resolution image. Restoration is again required to reduce the blurring introduced by

the video camera sensors.

Chapter 4 describes the high-resolution reconstruction algorithm. In addition, a comparison is made between images enhanced by single frame interpolation and multi-frame resolution enhancement. The evaluation is of algorithm performance, algorithm limitations, and computational intensity. Determining algorithm performance is done by inspecting the resulting images looking at high frequency details, edges, and overall image appearance. It is important that the image is not blurred as a result of processing since that will defeat the purpose of resolution enhancement. Algorithm limitations are specified to define what types of images can be enhanced and what types of motions can be successfully estimated in the video resolution enhancement algorithm.

Chapter 2

Interpolation

Interpolation is used to fit a continuous curve between discrete sample points. In this manner, the number of sample points of an image can be increased by resampling the continuous function that is generated by interpolation. Increasing the number of sample points in an image will result in an image which appears better than the original. Several interpolation methods have been described throughout the signal processing literature. This chapter will evaluate a few inexpensive interpolation methods.

When the size of an image is doubled in both directions, the number of sample points in the image is quadrupled. To quadruple the number of sample points in the original image, $\hat{g}(x, y)$, the image must first be expanded by 2 in each direction. i.e.:

$$\hat{f}(x,y) = \begin{cases} \hat{g}(x/2, y/2), & \textit{when both } x \textit{ and } y \textit{ are even} \\ 0, & \textit{otherwise} \end{cases} \tag{2.1}$$

This process quadruples the number of points in the image by inserting a zero in between the original sample points. The scaled image is completed by applying an interpolation method.

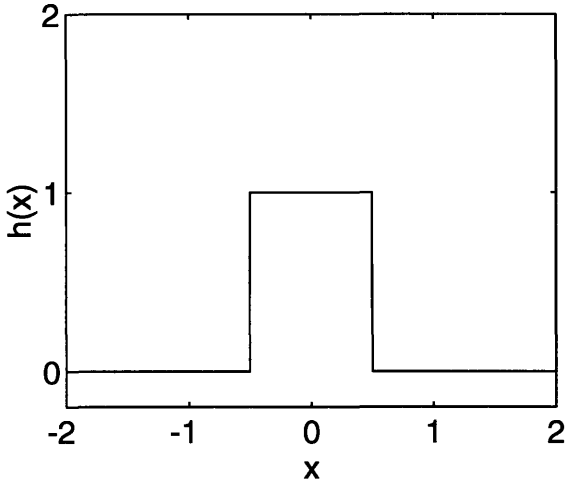

Figure 2-1: h(x) nearest-neighbor interpolation

2.1 Nearest-Neighbor Interpolation

A band-limited signal sampled at or above the Nyquist rate can be perfectly reconstructed (interpolated) by passing it through an ideal low pass filter [16]. The convolution kernel associated with the ideal low pass filter is the sinc function in the time domain. This sinc function is of infinite extent and can not be implemented.

The simplest way to scale an image, is to use nearest-neighbor interpolation [12, 16]. To increase the number of sample points, one simply replicates the original image sample points. Nearest-neighbor interpolation can be viewed as the convolution of a piecewise constant kernel with the original discrete sample points. The great advantage of using convolution kernels to do interpolation is that signal processing theory can be used to evaluate different methods. The continuous one dimensional kernel used for nearest-neighbor interpolation is shown in Figure 2-1. The kernel $h(x) = 1$ for $-0.5 \leq x < 0.5$. The original signal is defined for the integer values of x. The extension to two dimensions is straightforward: $h(x, y) = h(x)h(y)$.

The frequency response magnitude of the nearest-neighbor interpolation filter in one dimension is shown in Figure 2-2. The frequency response of the ideal low pass filter is shown for comparison. The first zero crossing of the nearest-neighbor interpolation filter extends beyond the cutoff frequency of the ideal low pass filter. As a result, this interpolation method has little resolution loss which is determined by the

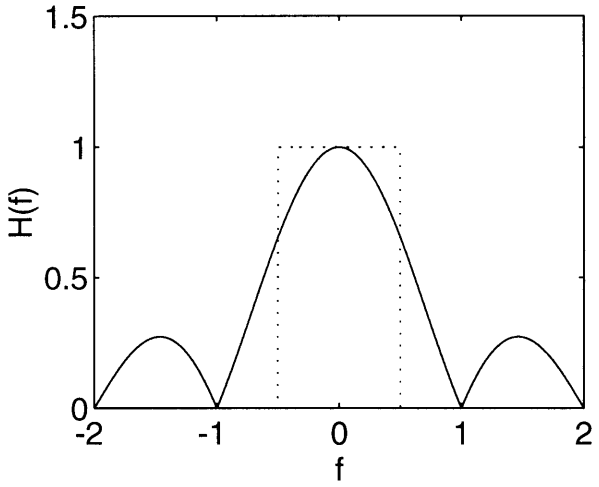

Figure 2-2: H(f) (nearest-neighbor interpolation —— , ideal low pass filter ....)

width of the main lobe of the interpolation filter. This is a desirable feature when the goal is resolution enhancement. Unfortunately, since this response has very high sidelobes the original signal must be very well sampled (beyond the Nyquist rate) such that false high frequencies from replicated spectrum are not preserved. This is known as interpolation error [12] and will result in noticeable pixel replication in the output image.

2.1.1 Example

The following is an example of nearest-neighbor interpolation. Figure 2-3 shows an original 200 by 200 image at full scale. After the image is expanded by 2 in each direction, the two dimensional interpolation filter is applied to the image. The scaled image is shown in Figure 2-4. The resulting output image has not been blurred significantly. However, interpolation error results in very noticeable jagged edges, since the original signal was not over-sampled.

2.2 Bilinear interpolation

Interpolation error can be reduced significantly by using a higher order interpolation filter. Linear interpolation (bilinear in two dimensions) applies a piecewise linear

Figure 2-3: Original image (200x200)

interpolation filter to the original sample points. The one dimensional convolution kernel for linear interpolation is shown in Figure 2-5.

The frequency response of this interpolation filter is shown in Figure 2-6. This filter has reduced sidelobes when compared with the nearest-neighbor interpolation filter. Therefore, noticeable jagged edges are not present when the signal is under-sampled. Unfortunately, this interpolation filter does more low pass filtering and the resulting output signal is slightly blurred. Therefore, there is a trade off between nearest-neighbor and the linear interpolation. The bilinear interpolation filter reduces jagged edges by smoothing the image. The result of bilinear interpolation is shown in Figure 2-7. The increase in resolution loss is apparent.

2.3 Cubic interpolation

Further reduction in interpolation error result when a higher order polynomial convolution kernel is applied. A common interpolation method used is a bicubic interpolation filter which consists of a product of 3rd order polynomials. Unfortunately, simple bicubic interpolation results in a severely smooth image. Figure 2-8 shows the result of simple bicubic interpolation. However, good results have been produced by using a finite support piecewise cubic convolution kernel.

Figure 2-4: Original image scaled using nearest-neighbor interpolation method (400x400)

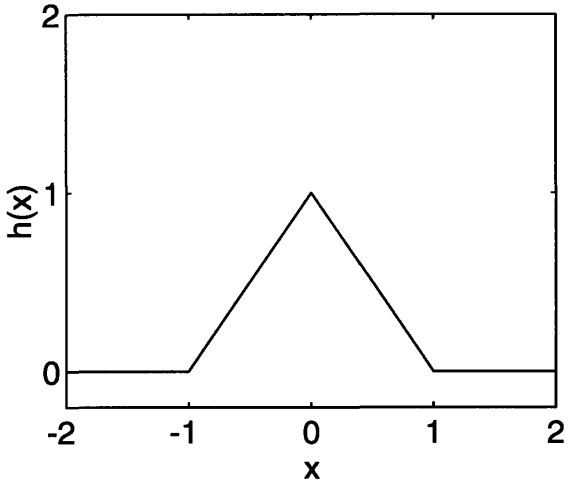

Figure 2-5: h(x) linear interpolation

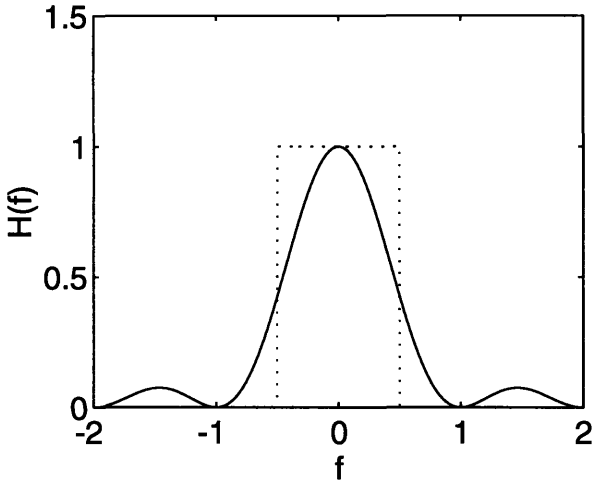

Figure 2-6: H(f) (linear interpolation —— , ideal low pass filter ....)

Figure 2-7: Original image scaled using bilinear interpolation method (400x400)

Figure 2-8: Original image scaled using bicubic interpolation method (400x400)

2.3.1 Piecewise Cubic Convolution

This section describes a method which uses a constrained piecewise cubic convolution kernel to interpolate a signal. This convolution kernel has been described to be an efficient approximation to the ideal reconstruction sinc function [19].

Good results have been produced by using a finite support piecewise cubic convolution kernel [19]. The one dimensional kernel, $h(x)$, is composed of piecewise cubic polynomials on the subintervals: $(-2,-1),(-1,0),(0,1),(1,2)$. The original signal is defined for the integer values of x. The extent of the filter is such that each interpolated point is derived by using 4 original sample points.

A piecewise one dimensional cubic convolution kernel can be written as follows:

$$h(x) = \begin{cases} a|x|^3 + b|x|^2 + c|x| + d, & 0 \leq |x| < 1 \\ e|x|^3 + f|x|^2 + g|x| + h, & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \tag{2.2}$$

This kernel is designed to be symmetrical so that it will have zero phase. Constraints are imposed on the kernel to determine the value of the coefficients. The first constraint is that $h(0) = 1$ and $h(n) = 0$, where n equals $-2, -1, 1$, and 2. This constraint implies that the interpolated continuous function will pass through the original sample points. Additional constraints are that $h(x)$ and its first derivative must be continuous at the boundary points. This will assure that the final interpolated function is continuous and smooth.

The constraints described above lead to seven equations and do not completely determine all eight coefficients for $h(x)$. There is a free parameter which can be used to adjust the filter as desired. Allowing e to be the free parameter results in the equations below.

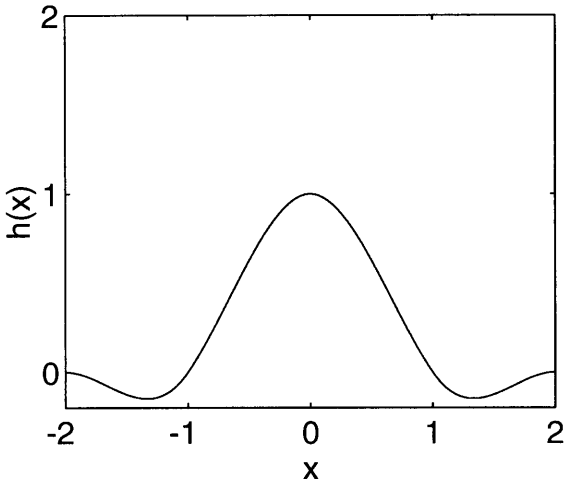

Figure 2-9: h(x) piecewise cubic interpolation

$$h(x) = \begin{cases} (e+2)|x|^3 - (e+3)|x|^2 + 1, & 0 \leq |x| < 1 \\ e|x|^3 - 5e|x|^2 + 8e|x| - 4e, & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \tag{2.3}$$

Several heuristics have been used to determine this free parameter. Since $h(x)$ is supposed to approximate the sinc function, e is constrained to be between -3 and 0. This assures that $h(x)$ is concave down at $x = 0$ and concave up at $x = 1, -1$. The following values of e have been suggested in literature: $e = -1$, $e = -.75$, and $e = -.5$. Setting e to -1 matches the slope of $h(x)$ to the sinc function at $x = 1, -1$. This choice of e is preferred for visual enhancement of the image. The frequency response of this filter amplifies frequencies in the high frequency part of the pass band. As a result, the image is sharpened. Setting $e = -.75$ allows the second derivative of $h(x)$ to be continuous at $x = -1, 1$. Finally, $e = -.5$ has been used for mathematical preciseness in that the Taylor series approximation of the interpolated function matches the original signal in as many terms as possible. [19].

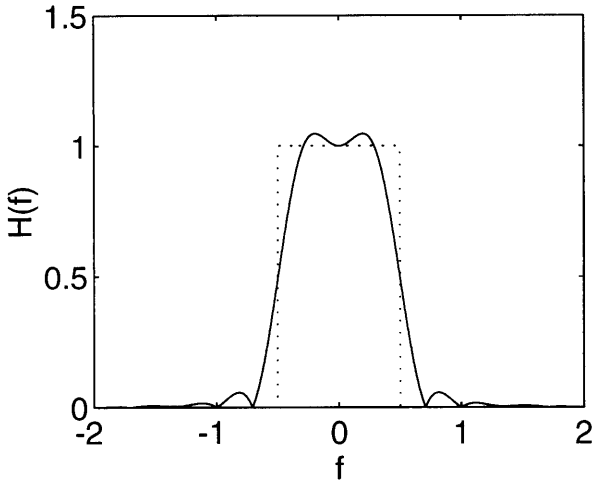

Figure 2-10: H(f) (piecewise cubic —— , ideal low pass filter ....)

2.3.2 Results

In the example that follows, the value $e = -1$ was used. Figure 2-9 shows the one dimensional version of the convolution kernel $h(x)$. Notice the resemblance to the sinc function. The one dimensional frequency response of the digital filter used to quadruple the number of sample points is shown in Figure 2-10 along with the ideal filter response.

After the image is expanded by 2 in each direction, the sampled, 7x7, two dimensional interpolating convolution kernel $h(x, y)$, is convolved with the expanded image. The origin of $h(x, y)$, shown in Figure 2-11, is the 1.0000 in the center of the figure. Since $h(x, y)$ is designed so that the output function will pass through the original sample points, these points do not need to be recalculated. The only output points that need to be calculated are the points where zeros were inserted to form $\hat{f}(x, y)$. Since $h(x, y)$ is symmetrical, the output image can be computed with an average of 5.25 floating point additions per pixel and an average of 1.75 floating point multiplications per pixel.

The resulting printed image is shown in Figure 2-12. This image is free from the artifacts present in the other scaled image.

0.0156	0.000	-0.0781	-0.1250	-0.0781	0.000	0.0156
0.000	0.000	0.000	0.000	0.000	0.000	0.000
-0.0781	0.000	0.3906	0.6250	0.3906	0.000	-0.0781
-0.1250	0.000	0.6250	1.0000	0.6250	0.000	-0.1250
-0.0781	0.000	0.3906	0.6250	0.3906	0.000	-0.0781
0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.0156	0.000	-0.0781	-0.1250	-0.0781	0.000	0.0156

Figure 2-11: $h(x, y)$

Figure 2-12: Scaled image using piecewise cubic interpolation method (400x400)

Chapter 3

Multiple Motion Estimation

In the previous chapter, interpolation was used to increase the number of sample points in an image. The resolution of the image is not enhanced, since no additional information is added to the image. If one compares the spectral contents of an image before and after interpolation, the information is identical.

To increase resolution and detail of objects in an image, more information from the scene must be gathered and correctly incorporated into the image to be enhanced. With motion, this may be accomplished using multiple frames from a video sequence. The precise sub-pixel displacement of pixels in successive frames must be determined, such that the addition of new information will successfully result in resolution enhancement.

Motion estimation between two frames can provide the required sub-pixel displacement. Two dimensional motion estimation assumes that the motion of each pixel from one frame to the next can be described as a pixel translation in the image plane. When motion estimation is done between frames, the translational velocity of each pixel in the first frame determines its sub-pixel displacement and its location in the second frame. Thus, new information can be used to increase resolution. However, it is not a simple requirement to provide accurate motion estimation from scenes with multiple motions.

3.1 Motion Estimation Overview

Many algorithms for motion estimation have been proposed throughout the image processing literature [1, 2, 3, 7, 11, 17]. This chapter describes and examines different motion estimation algorithms to determine which is most appropriate for video resolution enhancement with multiple motions. A consideration in evaluating a motion estimation algorithm is its accuracy. In addition, algorithm complexity must be considered. Since results of video resolution enhancement will be compared with interpolation, there is a trade off that must occur. Increases in computational intensity should result in improvement in reliability of motion estimates.

3.1.1 Block Matching

One simplifying assumption used in motion estimation is data conservation between frames (i.e. all objects in the first frame are located in the second frame at shifted positions)[1]. With $I(x, y, t)$ representing the image intensity I at point (x,y) and time t, data conservation implies for some horizontal and vertical velocity u and v:

$$I(x, y, t) - I(x + u, y + v, t + 1) = 0 \tag{3.1}$$

Block matching is the most intuitive approach to motion estimation between two frames. In a block matching algorithm, a region from frame 1 (usually an NxN block) is searched for in frame 2. At the correct motion estimate for each block, the difference in image intensity between the two frames should be equal to 0. To ensure reliable velocity estimates, a full search of frame 2 to find the match with a block from the first frame can be done.

In real situations, it is difficult to find a velocity estimate which can satisfy equation 3.1 for every pixel in the block. Consequently, the resulting velocity estimate of each block is determined by minimizing the following data conservation error equation [4]:

$$E_D(u,v) = \sum_{(x,y)\epsilon Block} [I(x,y,t) - I(x+u, y+v, t+1)]^2 \tag{3.2}$$

The least-square velocity estimates u and v for each block are assigned to each pixel in the block. Problems will arise when there are multiple motions in the block and when there is significant non-translational motion in the picture. With multiple motions, large block sizes may cross motion boundaries. The accurate motion of some pixels within the block are different from others, but they are all forced to have identical motion estimates.

In addition, since the desired motion estimate between frames must be of sub-pixel accuracy, the sampled image frame must be interpolated, and additional searching done once the initial integer velocity estimates are determined.

3.1.2 Region-Based Motion Estimation

To avoid the explicit search and interpolation, a continuous model of the sampled intensity image was assumed and a Taylor series expansion of the data conservation equation 3.1 above was applied [9]. Further, an assumption of small motion between frames allows the higher order terms to be dropped resulting in the familiar optical flow constraint equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{3.3}$$

This equation constrains the motion estimate to a line in u - v space. Since in real situations, it is difficult to find velocity estimates that satisfy equation 3.3 for every pixel in the region, the velocity estimate for the region is again determined by minimizing the following error equation:

$$E_D(u,v) = \sum_{(x,y)\epsilon Region} [\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}]^2 \tag{3.4}$$

An additional advantage to this new formulation of the data conservation equation is that a parametric motion model may be applied to the region. When a motion model is used, the velocities (u and v) are no longer restricted to a constant value for the whole region. Instead, each velocity to be estimated is determined by a parametric equation. The velocities in equation 3.4 are replaced by $u(x,y;\boldsymbol{a})$ and $v(x,y;\boldsymbol{a})$, where the motion model parameters to be estimated are represented by the variable $\boldsymbol{a}$ [3].

One popular model used is an affine flow model which uses 6 parameters to estimate global motion over a region. This motion model allows for translation, as well as rotation. Unfortunately, these models, which assume global motion, suffer from what is referred to as the generalized aperture problem, since the size of the region must be both large to constraint the solution and small to assure a single motion in the region [3].

A fundamental problem in motion estimation over a region which contains multiple motions is the interdependence between segmentation and motion estimation (Figure 3-1). To accomplish accurate motion estimation, the image frame must first be segmented into regions containing a single motion. However, correct segmentation of an image first requires accurate motion estimation. Simultaneous segmentation and motion estimation is required to assure reliable estimation.

A claim often made of region-based motion estimation is that the accurate motion of multiple objects in the region can be correctly estimated. When iteratively applied

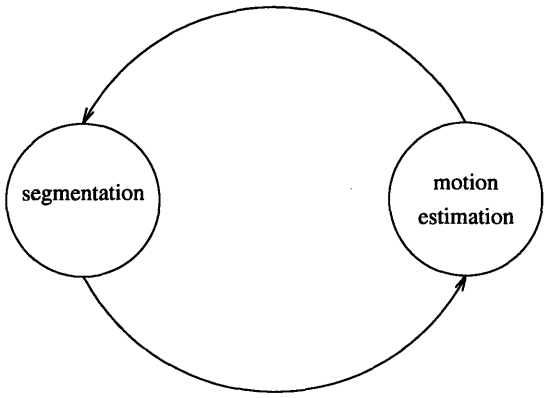

Figure 3-1: Interdependence between segmentation and motion estimation.

over a region containing multiple moving objects, the motion of the "dominant object" is selected [2, 17, 11, 7]. These approaches begin with no initial segmentation and iteratively estimate motion and then segment the region. Unfortunately, it is not clear what constitutes a "dominant object" in the region. In addition, convergence requires sufficiently large motion differences between objects [7]. In the absence of accurate segmentation, solutions to region-based motion estimation become unreliable, and region-based algorithms are not sufficient for multiple motion estimation.

3.1.3 Pixel-Based Motion Estimation

If the size of the region is reduced to a single point, the aperture problem can be avoided. Motion is estimated for each individual pixel. Let $u_{(x,y)}$ and $v_{(x,y)}$ be the velocity estimates at the point (x, y). The data constraint error equation to be minimized at each point becomes:

$$E_D(u_{(x,y)}, v_{(x,y)}) = [\frac{\partial I}{\partial x}u_{(x,y)} + \frac{\partial I}{\partial y}v_{(x,y)} + \frac{\partial I}{\partial t}]^2 \tag{3.5}$$

When motion parameters are estimated for each individual pixel, differently mov-

ing objects are no longer forced to have identical motion estimates. Unfortunately, the solution becomes under-constrained, since only information (data conservation) at a single point is used to solve for two dimensional motion $u_{(x,y)}$ and $v_{(x,y)}$.

To help constrain the solution, a smoothness constraint equation is added to the data conservation error equation [9]. One assumption made is that motion velocities vary smoothly over the image frame. With rigid body motion, neighboring pixels have similar velocities. A first order measurement of smoothness violation is the square of the velocity gradient magnitude:

$$(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 \text{ and } (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2 \tag{3.6}$$

When the motion parameters estimated are smooth over the image frame, the velocity gradient magnitudes are small. Thus, it can be used to form a smoothness error equation which is minimized at the correct estimated solution. Since the pixel sites are located on a discrete grid, velocity gradients must be estimated. A good approximation to the partial derivatives in equation 3.6 at point (x,y) is velocity differences over a small local neighborhood $\Omega_{(x,y)}$[3]. Thus, the smoothness error equation can be expressed as:

$$E_S(u_{(x,y)}, v_{(x,y)}) = \sum_{(i,j)\epsilon\Omega_{(x,y)}} [(u_{(x,y)} - u_{(i,j)})^2 + (v_{(x,y)} - v_{(i,j)})^2] \tag{3.7}$$

A first order neighborhood system $\Omega_{(x,y)}$ is defined to be the set of north, south, east, and west neighbors of a point (x,y). (Figure 3-2)

The total error equation to be minimized is formed by a weighted sum over the image frame of the data conservation and smoothness error terms at each point. Since the velocities across the image are assumed to be continuous and smooth, a membrane

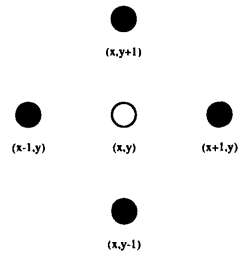

Figure 3-2: Neighborhood $\Omega_{(x,y)}$

[5] model of motion is formed.

$$E = \sum_{(x,y)} [\lambda E_D(u_{(x,y)}, v_{(x,y)}) + E_S(u_{(x,y)}, v_{(x,y)})] \tag{3.8}$$

λ is used to control the relative importance of velocity estimates that minimize the smoothness error term versus estimates that adhere to the data constraint equation. The resulting velocity is found between solutions that minimize each term separately. A least-squares formulation of the smoothness term alone is minimized when the velocity estimates are equal to the mean of the neighborhood velocities. This is shown by setting the derivatives of the smoothness error equation to zero and solving for $u_{(x,y)}$ and $v_{(x,y)}$.

$$\frac{\partial E_S}{\partial u_{(x,y)}} = 4u_{(x,y)} - \sum_{(i,j)\epsilon\Omega_{(x,y)}} u_{(i,j)} = 0 \tag{3.9}$$

$$\frac{\partial E_S}{\partial v_{(x,y)}} = 4v_{(x,y)} - \sum_{(i,j)\epsilon\Omega_{(x,y)}} v_{(i,j)} = 0 \tag{3.10}$$

As λ approaches zero, the estimated values of $u_{(x,y)}$ and $v_{(x,y)}$ approach the mean values, $\overline{u}$ and $\overline{v}$, over the neighborhood. This value of λ is appropriate only for constant velocity over the entire image frame. Setting λ too small will severely oversmooth the velocity estimates.

In addition, λ can not be set too high. At each point, the data conservation term is mimimized when the estimated values $u_{(x,y)}$ and $v_{(x,y)}$ satisfy the linear equation 3.3 at each point. However, the resulting motion estimates will be noisy due to expected errors in the estimation of partial derivatives [9].

Unfortunately, setting λ low enough to smooth the noisy results from the data conservation term may resust in oversmoothed results. This is best illustrated in Figure 3-3, which shows the constraints of the error equation 3.8 at a point (x, y) in $u-v$ space, namely the data conservation constraint line for the term E_D in equation 3.8 (see also equation 3.3) and the velocity estimates of 4 neighboring points N in the term E_S. One neighboring point belongs to a separate independently moving object. Hower, the estimation algorithm pools information, and a velocity estimate that best satisfies all of the terms in the error equation is determined. The estimated velocity will fall between the cluster of neighboring estimates and the separate neighbor estimate. Therefore, the incoherent neighbor will pull the answer away from the correct motion estimate.

The error equation as it is stated in equation 3.8 works well when the membrane model of motion is true. However, multiple moving objects in an image sequence give rise to motion discontinuities between regions. In these video sequences, motion is not smooth; it is piecewise-smooth.

3.1.4 Robust Estimation

A way to smooth the motion estimates while preserving motion discontinuities is needed. The pixel-based motion estimator needs to be made robust such that it does not penalyze for violations in symplifying assumption. The influence, on the estimation process, of neighboring points with incoherent motion needs to be eliminated or reduced. Two slight modifications to the pixel-based motion estimation

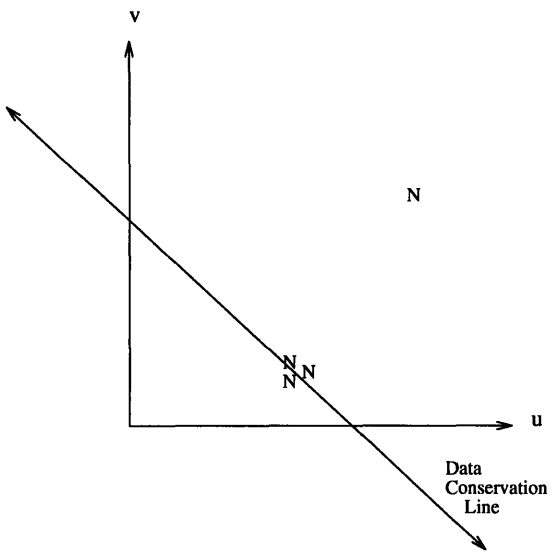

Figure 3-3: Error constraints in $u - v$ space

algorithm have been proposed which provide robust estimation of motion. They are the line-process and outlier-rejection.

Line-Process

A line process is a means of describing the location of motion discontinuties between pixels in an image frame. If these locations were known a priori, the smoothness constraint between two separetly moving neighbors could be disregarded when estimating motion. A binary line-process variable $l_{(x,y)(i,j)}$ is defined between points (x, y) and (i, j) as follows:

$$l_{(x,y)(i,j)} = \begin{cases} 1, & \textit{when motion discontinuity is present} \\ 0, & \textit{otherwise} \end{cases} \tag{3.11}$$

Line-process variables can be used in the pixel-based motion error equation. When a motion discontinuity is present between two neighboring pixels, the corresponding smoothness error term can be "turned off" by setting the line variable $l_{(x,y)(i,j)}$ be-

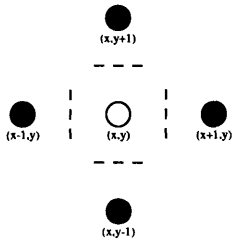

Figure 3-4: Neighborhood ($\Omega_{(x,y)}$) and possible discontinuities (- - -)

tween the neighboring points to 1, such that only information from coherently moving neighbors is used to determine the motion at that point. Figure 3-4 shows a neighborhood and possible motion discontinuities.

Unfortunately, motion discontinuities are not known a priori, and the values of the line process $l_{(x,y)(i,j)}$ must be estimated simultaneously with the motion estimation. This leads to a very complex minimization process, since the number of variables to be estimated has increased. The modified error equation 3.8 becomes:

$$E = \sum_{(x,y)} [\lambda E_D(u_{(x,y)}, v_{(x,y)}) + \tag{3.12}$$

$$\sum_{(i,j)\epsilon\Omega_{(x,y)}} [(1 - l_{(x,y)(i,j)})((u_{(x,y)} - u_{(i,j)})^2 + v_{(x,y)} - v_{(i,j)})^2) + \beta l_{(x,y)(i,j)}]] \tag{3.13}$$

A penalty term, $\beta l_{(x,y)(i,j)}$ has been added to discourage the introduction of lines. Motion discontinuity is a rare occurrence in most motion sequences. Without the penalty term, the estimation process will tend to over-segment the image, and lines

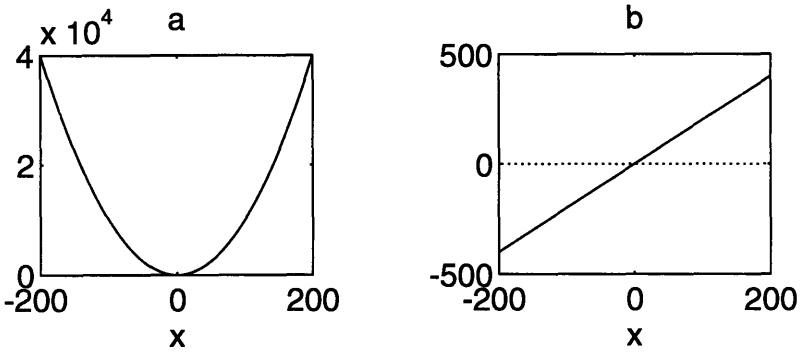

Figure 3-5: (a) $\rho(x) = x^2$, (b) $\psi(x) = 2x$

will be placed everywhere.

The error equation to be minimized is now non-convex and has many local minima. The non-convex nature is seen as follows: for a specific estimate of the line variables $l_{(x,y)(i,j)}$, there is a certain velocity estimate which minimizes the error equation. Different estimates of the line variables $l_{(x,y)(i,j)}$ will produce a different velocity estimate which locally minimizes the error equation but is not necessarily the global minimum. The problem of minimizing non-convex energy equations has been studied extensively and stochastic minimization techniques, such as simulated annealing have been developed to find the global minimum [8]. However, these processes are computationally expensive.

Outlier-Rejection

Outliers are points where the simplifying assumptions of data conservation and motion smoothness in video sequences are violated. The point-based estimator above assumed a membrane or continuous model of velocities over the image frame. Therefore, the resulting motion estimates smoothed over motion discontinuities. Neighboring points belonging to separate moving objects are outliers with respect to the smoothness constraint assumption. To estimate the correct motion, the influence of outliers must be reduced or eliminated (as in the line-process). The membrane motion model must become weak[5], in that occasional discontinuities (breaks in the smoothness assumption) must be allowed.

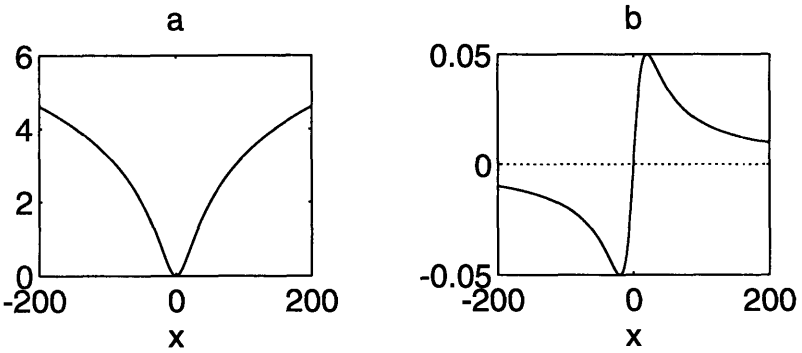

Figure 3-6: (a) $\rho(x,\sigma) = \log(1 + \frac{1}{2}(\frac{x}{\sigma})^2)$, (b) $\psi(x,\sigma) = \frac{2x}{2\sigma^2+x^2}$

The pixel-based motion estimation algorithm uses a quadratic estimator. When outliers are present, their resulting arguments (velocity difference) in the smoothness term are high at the correct velocity estimate. In this respect, the quadratic function is not a good estimator for a weak membrane model of motion. This can be seen by looking at its derivative $\psi(x)$ in Figure 3-5 [1]. $\psi(x)$ is termed as its influence function, since it determines how much the value of the estimator changes with a change of argument [3]. At the correct motion estimate, coherent neighbors have low arguments in the smoothness terms, since the velocity differences will be small. As a result, their influence is low relative to that of outlying neighbors. When the velocity estimates move away from the correct solution toward the outlier velocity, the decrease in the error equation from the outlier terms is greater then the increase from terms of coherent neighbors. Therefore, when outliers are present the motion estimate will be pulled away from the correct solution. Thus, the result is an over-smoothed motion estimate.

A robust estimator is used in [3] which does not use a line-process to estimate motion. The advantage of using a robust estimator is that outliers can be rejected automatically, and the computational expense of an explicit line-process is reduced. A Lorentzian function is used (Figure 3-6) instead of a quadratic estimator. The influence function $\psi(x,\sigma)$ begins to decrease after a certain argument value ($x = \tau$) called the outlier threshold . This point is determined by the control parameter

[1] Following the notion of [3], $\rho(x)$ is the estimator function and $\psi(x)$ is its derivative.

($\tau = \sigma\sqrt{2}$). Beyond this point, outliers begin to have a lower influence on the estimation of velocity. This is what is desired for an estimator in the weak membrane model of motion. Motion estimates will still be smooth. However, when occasional discontinuities are present, outliers will have very small influence and will not over-smooth the results.

In addition to the smoothness error terms, the data conservation term is also made robust, since there are points where this assumption may be violated due to noise in the image and motion boundaries [3]. The new motion estimation error equation becomes (compare with equation 3.8):

$$E = \sum_{(x,y)} [\lambda(\rho(\frac{\partial I}{\partial x} u_{(x,y)} + \frac{\partial I}{\partial y} v_{(x,y)} + \frac{\partial I}{\partial t}, \sigma_1) + \sum_{(i,j)\epsilon\Omega_{(x,y)}} [\rho(u_{(x,y)} - u_{(i,j)}, \sigma_2) + \rho(v_{(x,y)} - v_{(i,j)}, \sigma_2)]] \tag{3.14}$$

The quadratic is replaced by the Lorentzian function $\rho(x, \sigma)$. This formulation is again non-convex. If the data conservation term at a point and any of the smoothness terms with the 4 neighbors disagree, any term may be treated as an outlier while the others are minimized. This results in many local minima. A common method used to find a local minimum is simultaneous over relaxation (SOR) which iteratively updates the current velocity estimates to minimize the error equation. At iteration $(n + 1)$ the update equations are [5]:

$${u_{(x,y)}}^{(n+1)} = {u_{(x,y)}}^{(n)} - w \frac{1}{T(u_{(x,y)})} \frac{\partial E}{\partial u_{(x,y)}} \tag{3.15}$$

$$v_{(x,y)}{}^{(n+1)} = v_{(x,y)}{}^{(n)} - w\frac{1}{T(v_{(x,y)})}\frac{\partial E}{\partial v_{(x,y)}} \tag{3.16}$$

where $0 < w < 2$ is the "SOR parameter" which controls the speed of convergence. $T(\cdot)$ is the upper bound of the second derivative of the error equation with respect to the velocity.

$$T(u_{(x,y)}) = \frac{\lambda(\frac{\partial I}{\partial x})^2}{{\sigma_1}^2} + \frac{4}{{\sigma_2}^2} \geq \frac{\partial^2 E}{\partial u_{(x,y)}{}^2} \tag{3.17}$$

$$T(v_{(x,y)}) = \frac{\lambda(\frac{\partial I}{\partial y})^2}{{\sigma_1}^2} + \frac{4}{{\sigma_2}^2} \geq \frac{\partial^2 E}{\partial v_{(x,y)}{}^2} \tag{3.18}$$

At each step, the current velocity estimates are corrected to reduce the error equation. The partial derivative of the error equation (shown for $u_{(x,y)}$ only) is [3]:

$$\frac{\partial E}{\partial u_{(x,y)}} = \lambda\frac{\partial I}{\partial x}[\psi(\frac{\partial I}{\partial x}u_{(x,y)} + \frac{\partial I}{\partial y}v_{(x,y)} + \frac{\partial I}{\partial t}, \sigma_1)] + \sum_{(i,j)\epsilon\Omega_{(x,y)}} \psi(u_{(x,y)} - u_{(i,j)}, \sigma_2) \tag{3.19}$$

The advantage of using a robust estimator is easily shown when minimizing using the SOR algorithm. The change at each step in magnitude and direction of the velocity at each point depends on the current value of the influence function $\psi(x, \sigma)$ of each of the 5 terms in the error equation. When the quadratic is used, outliers

incorrectly have a large influence. To make the membrane model weak, the influence of outlier terms must be diminished. This is accomplished when the Lorentzian estimator is used.

Graduated Non-convexity

As stated before, a local minimum can be found using the SOR algorithm. However, the global minimum of the non-convex error equation needs to be found. The computational expense of stochastic approaches discourages their use. Fortunately, the error function to be minimized is now differentiable and a more efficient minimization technique is used. Specifically, the graduated non-convexity (GNC) algorithm presented in [5] is used to minimize the error equation.

GNC minimizes the error function by first constructing a convex approximation to the error function which is easily minimized using the SOR algorithm described above. Successive approximations which slowly converge to the original error function are constructed. Once the local minimum is found at each GNC stage using the SOR algorithm, the resulting velocity estimates are used as the initial estimates for the next stage. Unfortunately this technique does not guarantee global optimization of the error function. A trade off is made: computational intensity is reduced, leaving the possibility of non-optimal results.

The GNC algorithm is readily applicable to the robust formulation with the Lorentzian as the estimator. The sequence of error functions to be minimized can easily be constructed by varying the control parameters σ_1 and σ_2. The value of σ_1 determines the outlier threshold τ_1 for the optical flow argument to the data conservation term in the error equation. The value of σ_2 determines when any of the spatial smoothness terms are considered outliers. By setting each parameter sufficiently high, the initial function is locally convex, since no term can be considered an outlier. The value of each control parameter is slowly lowered to construct the sequence of functions to be minimized until the final error function is reached. In this manner, outliers are slowly introduced [3].

Parameter Values

To construct the convex approximation, all argument values must be less than the outlier threshold τ. If the largest possible argument values are known or estimated, each σ could be set such that the outlier threshold is beyond these values.

The initial velocity estimates are set to 0 everywhere to begin the SOR iterations to minimize the convex function. Thus, the initial value of the data conservation argument $\frac{\partial I}{\partial x}u_{(x,y)} + \frac{\partial I}{\partial y}v_{(x,y)} + \frac{\partial I}{\partial t}$ is equal to $\frac{\partial I}{\partial t}$ everywhere. A conservative estimate for the outlier threshold τ_1 for the convex approximation is the $max|\frac{\partial I}{\partial t}|$. σ_1 to determine the convex approximation is set equal to this $\frac{\tau_1}{\sqrt{2}}$.

To determine the outlier threshold τ_2 for the convex approximation, we first guess the largest possible velocity v_L between the two frames. The estimate for τ_2 is set equal to $2v_L$ which will account for the largest possible velocity difference between two neighboring pixels. σ_2 for the convex approximation is set equal to this $\frac{\tau_2}{\sqrt{2}}$.

The final error function can be determined by estimating the outlier thresholds τ_1 and τ_2 for the final GNC stage and setting each control parameter σ_1 and σ_2 to $\frac{\tau_1}{\sqrt{2}}$ and $\frac{\tau_2}{\sqrt{2}}$ respectively. It remains a challenge to determine the optimal threshold value for the data and smoothness term for the final function. In general, for the smoothness term , it can be assumed at the final stage any small velocity difference is an outlier. In practice, many trials with different threshold values were attempted to find the best parameter values. Determining optimal parameter values remains a current drawback to the algorithm and requires further research.

Once the initial and final parameter values are determined, the sequence of functions are constructed using a fixed number of GNC stages. For simplicity, the parameters vary on a linear path form start to finish. They define the function to be minimized at each stage. The parameter λ is difficult to determine formally, and is empirically determined to be 10.

Experimental Results

To cope with large motions, a coarse-to-fine strategy is used based on the Laplacian pyramid structure described in [6]. A Laplacian pyramid structure separates an image

into band-passed filtered images of lower resolution. Empirical results show better motion estimation on bandpassed filtered images. Once an image is filtered, it is down sampled by 2. When the motion estimates have been determined at the image of lowest resolution (top of the pyramid), the motion estimates are projected to the next pyramid level and the image is warped according to the displacement estimates from the higher level. Motion estimation is done using the warped image. The residual motion calculated is added to the pre-warp velocities, and the process continues until the bottom of the pyramid is reached. The complete motion estimation method is describe in the block diagram in Figure 3-7. It differs slightly than that described in [3]. The full GNC minimization (all stages) is done at each pyramid level before moving to the next level.

An experiment of the motion estimation algorithm is done using computer generated images. Figure 3-8 shows a random textured background with 2 random textured blocks. The top block moves 1 pixel to the left, while the bottom block moves 1 pixel to the right. The background is stationary. Four GNC stages for each level in the pyramid are constructed with the initial parameter values determined as described above. The final σ_1 and σ_2 values are set to 20 and .09 respectively. At each stage 200 iterations of the SOR algorithm is done, where a single iteration is the updating of $u_{(x,y)}$ and $v_{(x,y)}$ at every point in the image.

Motion discontinuities can be recovered from the final velocity flow field. These are points in which the final velocity difference between any neighbor exceeds the outlier threshold τ_2 for the smoothness term in the final error function minimized. Figure 3-9 shows the detected motion discontinuities for the computer generated motion example.

This outlier-rejection algorithm provides the benefit of a motion estimation algorithm which is robust enough to estimate motion in image scenes with multiple moving objects. In addition, since the error function to be minimized is differentiable and can be solved using the simpler GNC method, the computational intensity is reduced when compared to the more intense simulated annealing method required to estimate motion with the line-process.

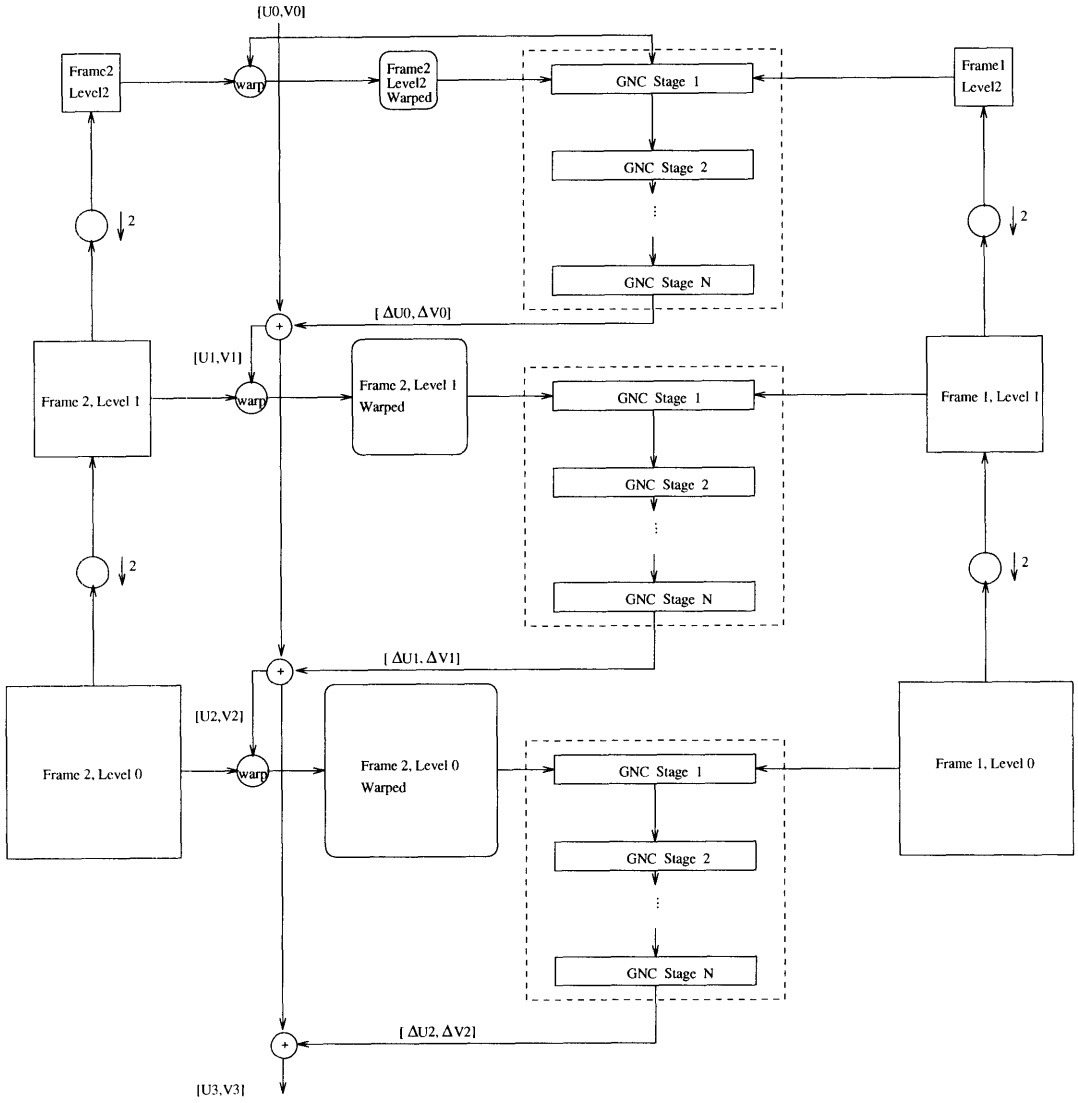

Figure 3-7: Motion estimation algorithm

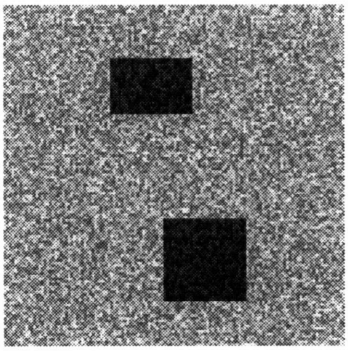

Figure 3-8: Computer generated frame

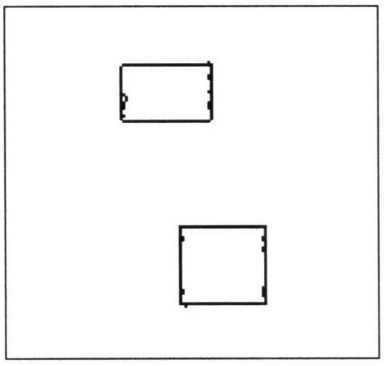

Figure 3-9: Motion discontinuities

Chapter 4

High-Resolution Reconstruction

4.1 High-Resolution Reconstruction

A motion estimation algorithm has been established, and the relative pixel displacements between the reference frame and successive frames can be determined. This chapter describes a method to reconstruct the high-resolution image and compares the results with interpolation.

The discrete high-resolution image of the original scene $\hat{f}(x,y)$ is reconstructed by using the original reference frame and adding information from successive frames. First, to initialize the high-resolution image, the low-resolution image is expanded to the size of the high-resolution image. For example, to double the number of sample points in each direction, the high-resolution image is initialized as follows:

$$\hat{f}(x,y) = \begin{cases} \hat{g}(x/2, y/2), & \textit{when both } x \textit{ and } y \textit{ are even} \\ 0, & \textit{otherwise} \end{cases} \tag{4.1}$$

Figure 4-1 illustrates this process. The grid sites with filled circles in the high-resolution frame represents the expanded information from the reference frame. The additional sites must be filled by either interpolation or with information from suc-

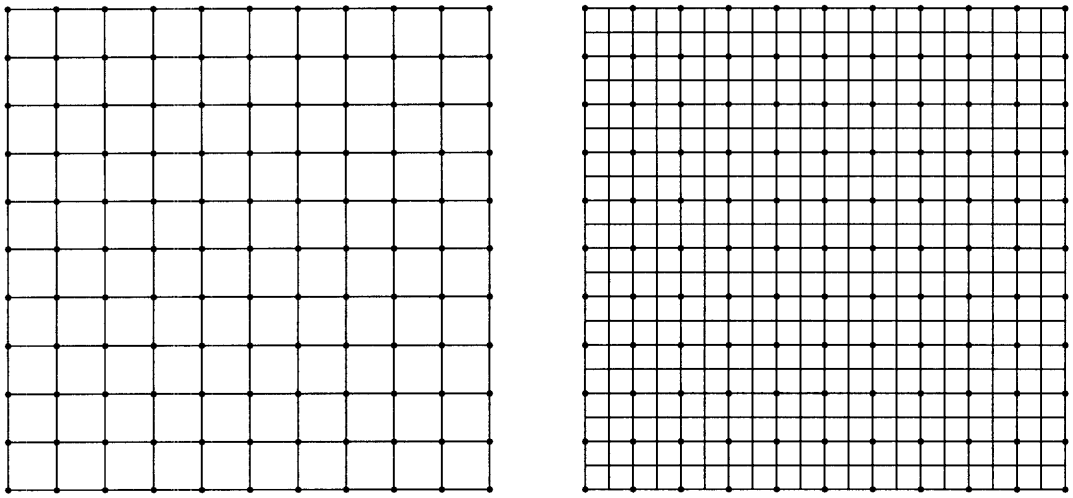

Figure 4-1: Low-resolution and initial high-resolution sampling grid

cessive frames which is provided by sub-pixel motion between frames.

Motion estimation is done between the original sampled frame and successive frames. Therefore, once the pixel displacement is estimated, the motion vector must be appropriately translated to finds its position in the high-resolution grid. When motion estimation is done between the reference frame and all other frames, the estimated location of every point, which can be any fractional displacement from a sampling location in the high-resolution grid, is known. Since there are several independently moving objects and several frames the resulting high-resolution image is non-uniformly sampled and interpolation must be done to display the high-resolution image on a uniform grid.

Interpolation of a non-uniformly sampled image to a uniform grid is a difficult problem. Any interpolation involves forming an initial assumption about the continuous function that is sampled. Since the goal of this reconstruction is to produce a high-resolution image, any process which blurs the output must be avoided. In this case, nearest-neighbor interpolation is used to go from the non-uniform to uniform sampling grid. Interpolation experiments in Chapter 2 concluded that any higher order interpolation technique will blur the image.

The high-resolution grid is filled with the nearest point to the desired high-

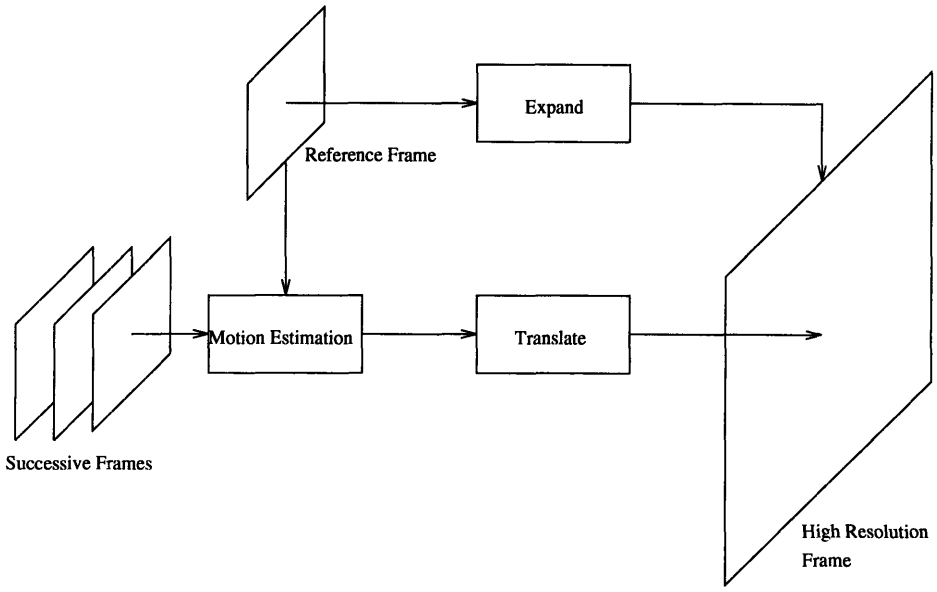

Figure 4-2: High-resolution reconstruction

resolution sampling grid location. It is assumed that the estimated fractional displacement of every pixel from all the frames will be unique. If it is not, the disputed location is filled with the first frame pixel to fill the location.

Nearest-neighbor interpolation, in general, may result in interpolation error and noticeable pixel replication in the output image. However, with sub-pixel motion in the sequence and several frames, it is assumed that the location of the nearest-neighbor will be close to the desired sampling position. In addition, noticeable pixel replication may be avoided, since the sample point added from a successive frames is not necessarily identical to the surrounding sample points. The total reconstruction process is shown in Figure 4-2. A reference frame is taken and combined with information from additional frames.

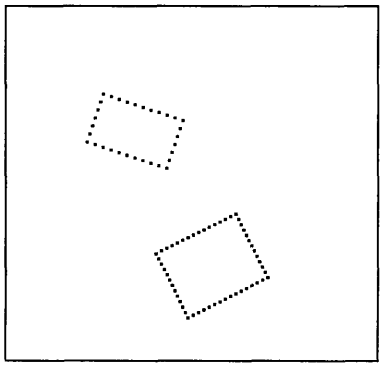

Figure 4-3: Object boundaries

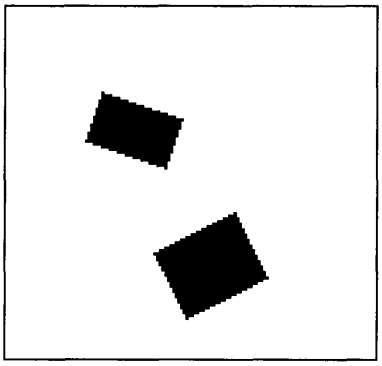

Figure 4-4: Synthetic computer image

4.2 Example of a Synthetic Image

Now, after we described the video resolution enhancement algorithm, a comparison can be made of this algorithm with different single frame interpolation methods. Results from the two processes can be evaluated to determine algorithm performance and potential limitations. The first example is from a computer generated image sequence. The synthetic images are created by direct sampling of 2 conceptual objects in the scene. Each object is a solid rectangle defined by lines through the boundaries shown in Figure 4-3.

The objects can not be sampled well enough and are poorly realized in the first image frame shown in Figure 4-4. The object boundaries are jagged. Any attempt to increase the resolution of the object by interpolation will surely fail. In the following examples, the size of the synthetic image is doubled in each direction.

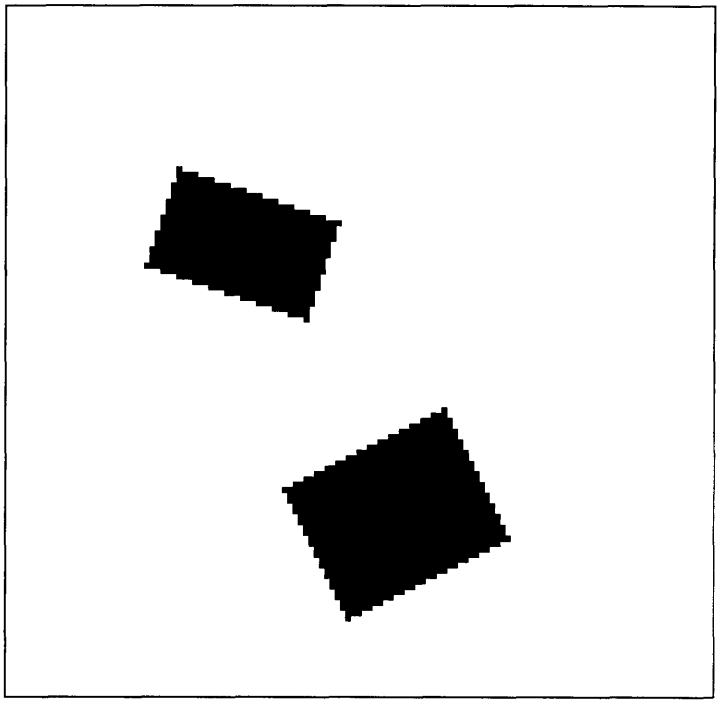

Figure 4-5: Nearest-neighbor interpolation

4.2.1 Interpolation

The synthetic image is first interpolated using nearest-neighbor interpolation (see Figure 4-5). Since the original image is under-sampled, pixel replication is evident in the final image. There is very little resolution loss. However, the resulting image from this interpolation has done a poor job in reconstructing a high-resolution image of the actual objects.

The results from higher order interpolation methods do a better job, since the gross interpolation error is reduced. However, the image of both rectangles is again poorly captured, since the resulting images are blurred. The results from bilinear and piecewise cubic interpolation are shown in Figures 4-6 and 4-7 respectively.

4.2.2 Multi-Frame Resolution Enhancement

Each interpolation method assumes an underlying model of the scene given the sample points of a single frame. The specific image model assumptions are evident in the resulting interpolated images. In video sequences with moving objects, sub-pixel mo-

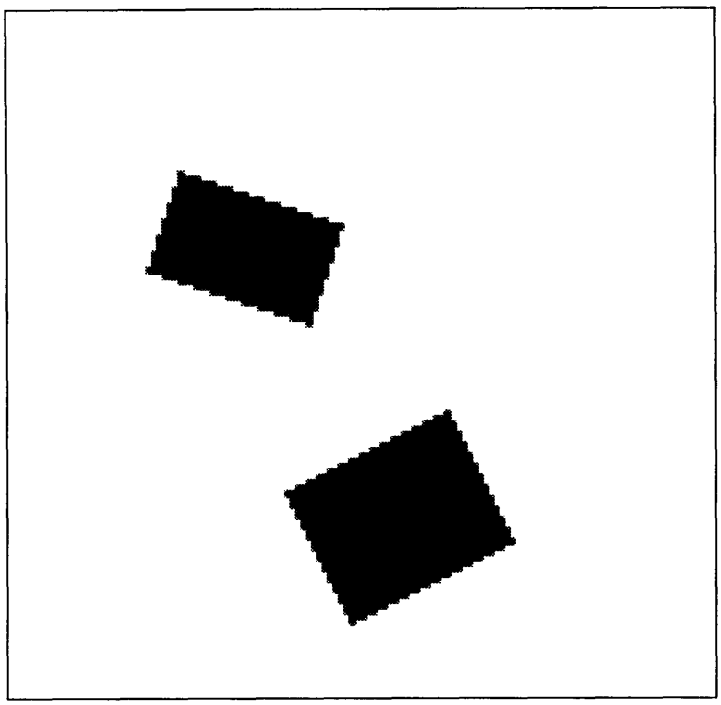

Figure 4-6: Bilinear interpolation

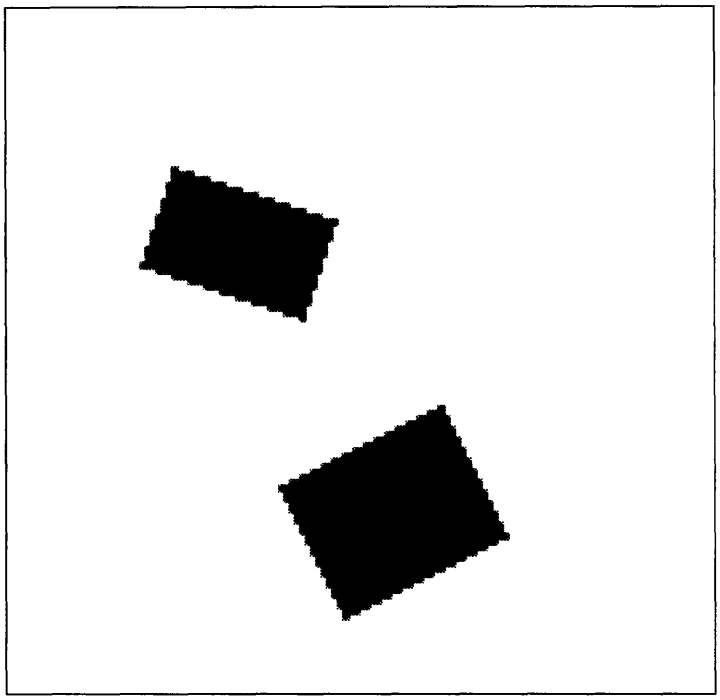

Figure 4-7: Piecewise cubic interpolation

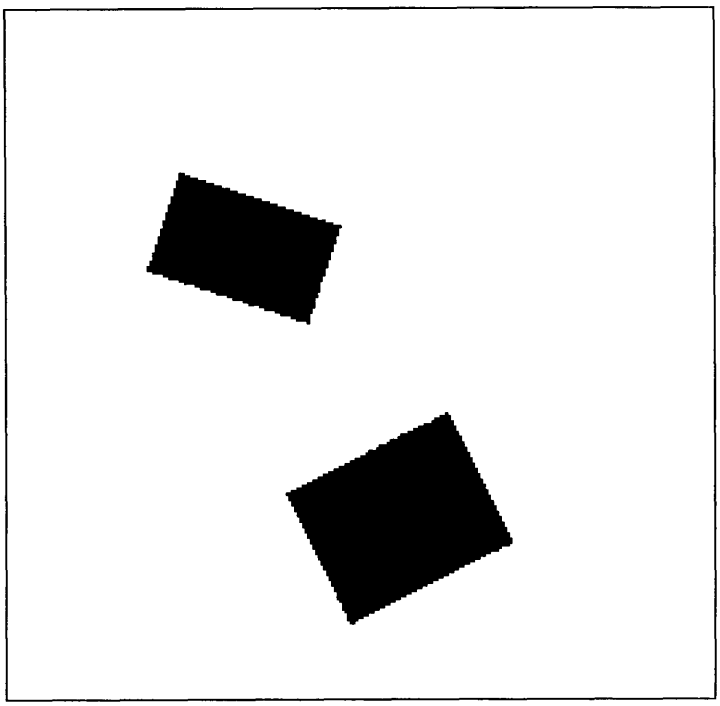

Figure 4-8: Multi-frame resolution enhancement

tion can be exploited to gain actual information from the image. With this additional information, incorrect scene models can be avoided.

Three additional frames of the objects are created by moving the rectangle objects half of a pixel in different directions with each frame and sampling the images. To create the first additional frame, the top rectangle moves to the left while the bottom moves oppositely to the right. Further frames are created with the top and bottom rectangles moving respectively towards the top and the bottom for the next frame and to the left and right for the final frame. The high-resolution reconstruction proceeds as described in the block diagram in Figure 4-2 with the original frame as the reference frame. The result is shown in Figure 4-8. Since actual scene information is used, this result is much closer to the underlying scene object.

4.3 Examples of Real Images

The example on synthetic images shows that the use of multiple frames has an advantage in resolution enhancement, since lost information can be restored. This section

-0.0111	-0.0111	-0.0111
-0.0111	1.0889	-0.0111
-0.0111	-0.0111	-0.0111

Figure 4-9: High pass filter $h(x, y)$

presents examples of resolution enhancement on real image scenes. The results using multiple image frames are compared with those from interpolation techniques.

4.3.1 Reduction of Image Blurring

The original images have poor resolution due to the blurring degradation caused by the camera sensors. Before resolution enhancement can be accomplished, this initial blur must be reduced. Image blur can be modeled as the convolution of the original image and a blurring function. To reduce this degradation, the blurring function is estimated for the camera, and inverse filtering is applied to each of the original degraded digital images $\hat{g}(x, y)$. For the following examples, the image blurring function is modeled with a Gaussian , and the images are high pass filtered before processing. The parameters for the 3x3 Gaussian blurring function are found in [6]. The coefficients for the high pass filter used are shown in Figure 4-9.

4.3.2 Two Motions

The simplest multiple motion example consists of a stationary background and a moving object. Figure 4-10 shows the original image to be enhanced [1]. The images are obtained from an NTSC video camera using a frame grabber. Motion for successive frames is produced by shifting the book while keeping the camera stationary.

The results from several interpolation methods described in Chapter 2 are shown in Figures 4-11, 4-12 , and 4-13. As expected, the interpolated images are slightly blurred.

[1] The image contains degradation seen as "ringing" artifacts from deficiencies in the frame grabber used to obtain the images used in this example.

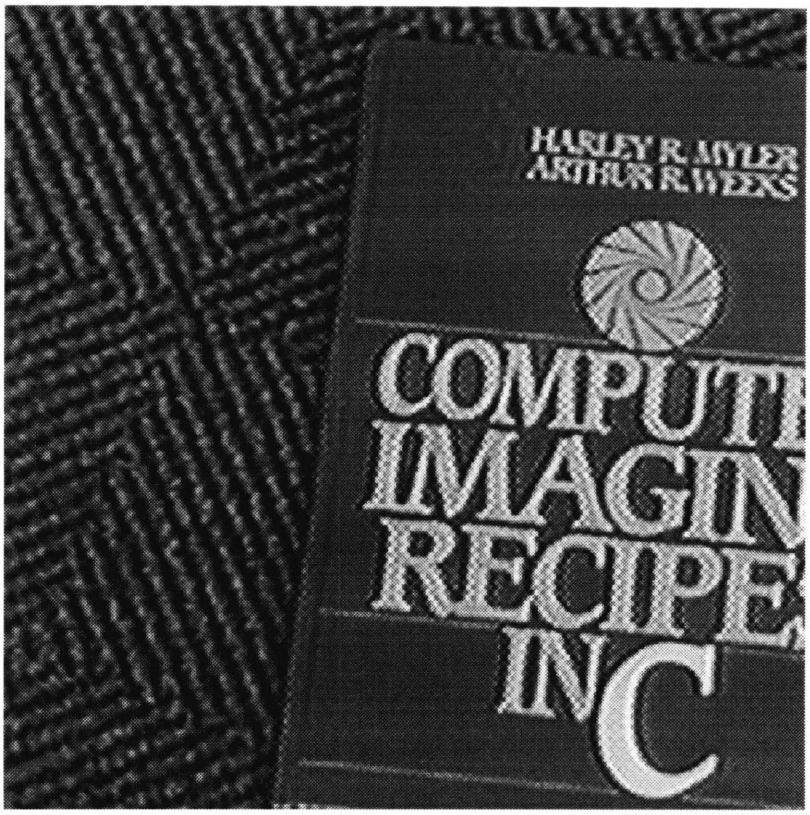

Figure 4-10: Original image frame

Figure 4-14 shows the results of multi-frame resolution enhancement using 4 video frames. This method produces a sharper image than interpolation. The benefit of using multiple frames algorithm is shown in figure 4-15 which compares a portion of the image obtained from the different interpolation techniques and the multi-frame resolution enhancement algorithm. The motion estimation is done using 3 pyramid levels. Only the final GNC stage is used which produces good motion estimates in this example.

4.3.3 Three Motions

The following example increases the number of motions in the image scene to determine possible limitations on multi-frame resolution enhancement. Figure 4-16 shows the original image taken with a digital camera. The examples from interpolation are shown in Figures 4-17, 4-18 , and 4-19.

The 2 objects are moved in opposite directions to create additional frames. A 4

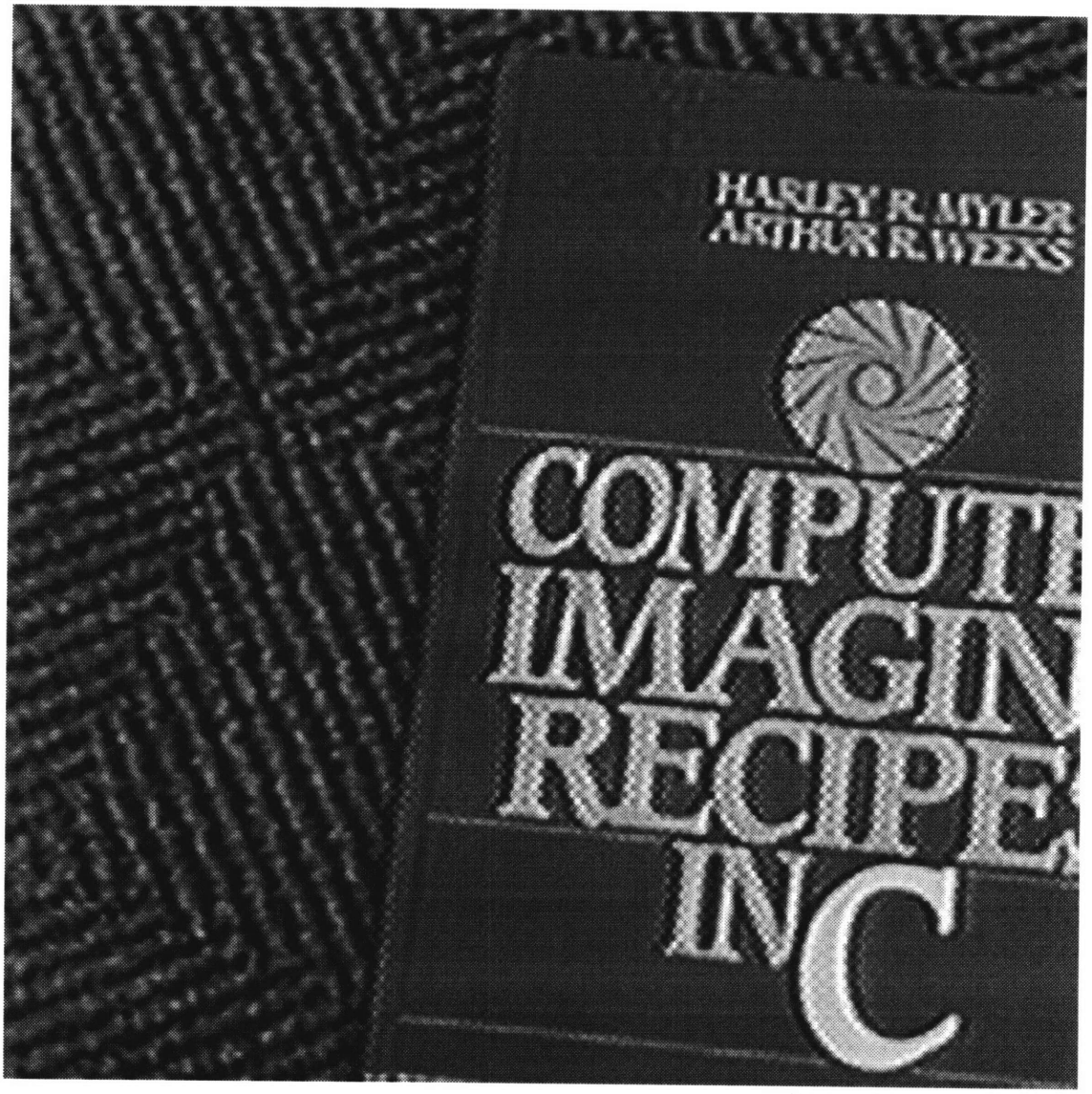

Figure 4-11: Bilinear interpolation

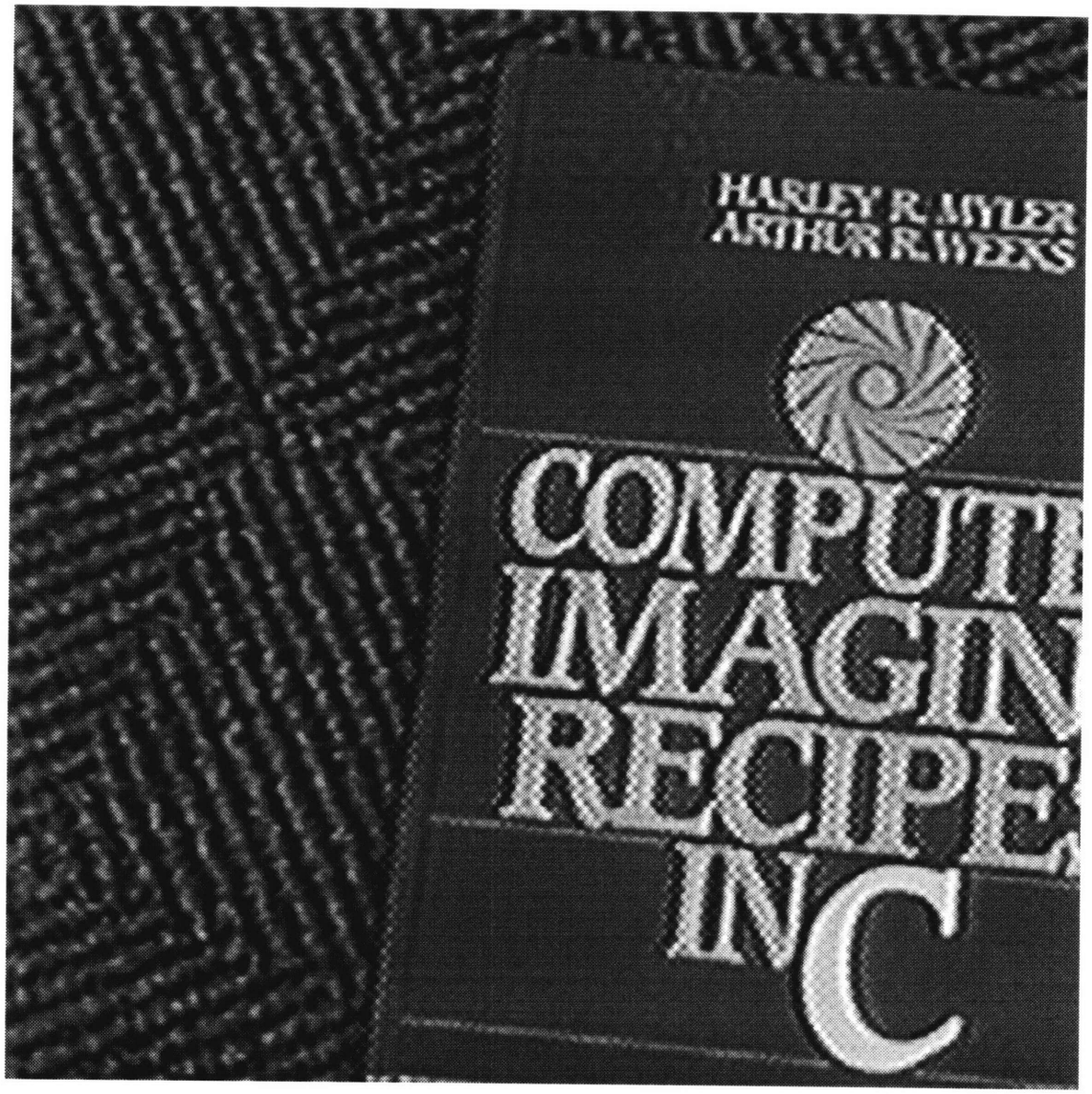

Figure 4-12: Bicubic interpolation

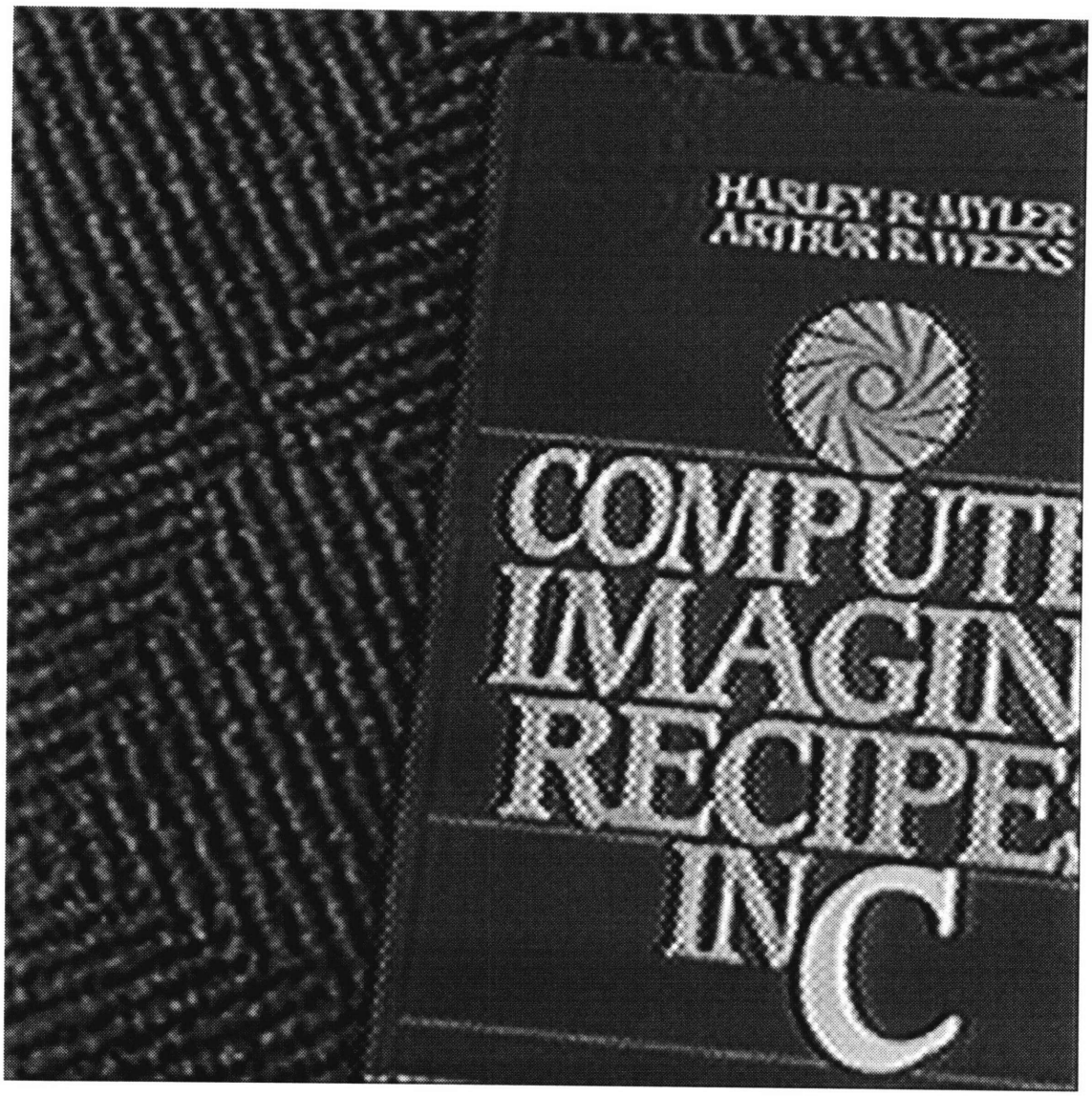

Figure 4-13: Piecewise cubic interpolation

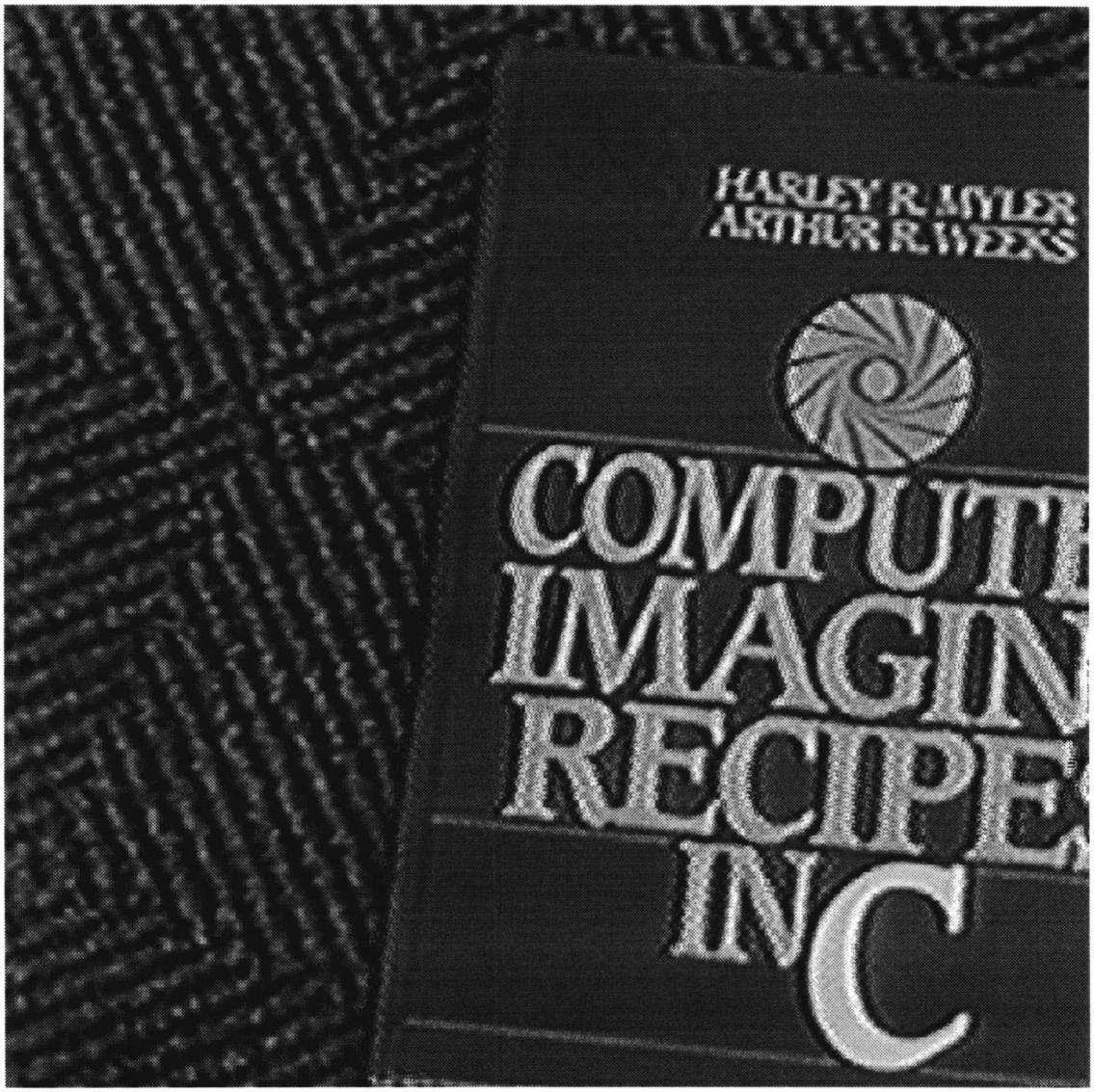

Figure 4-14: Multi-frame resolution enhancement

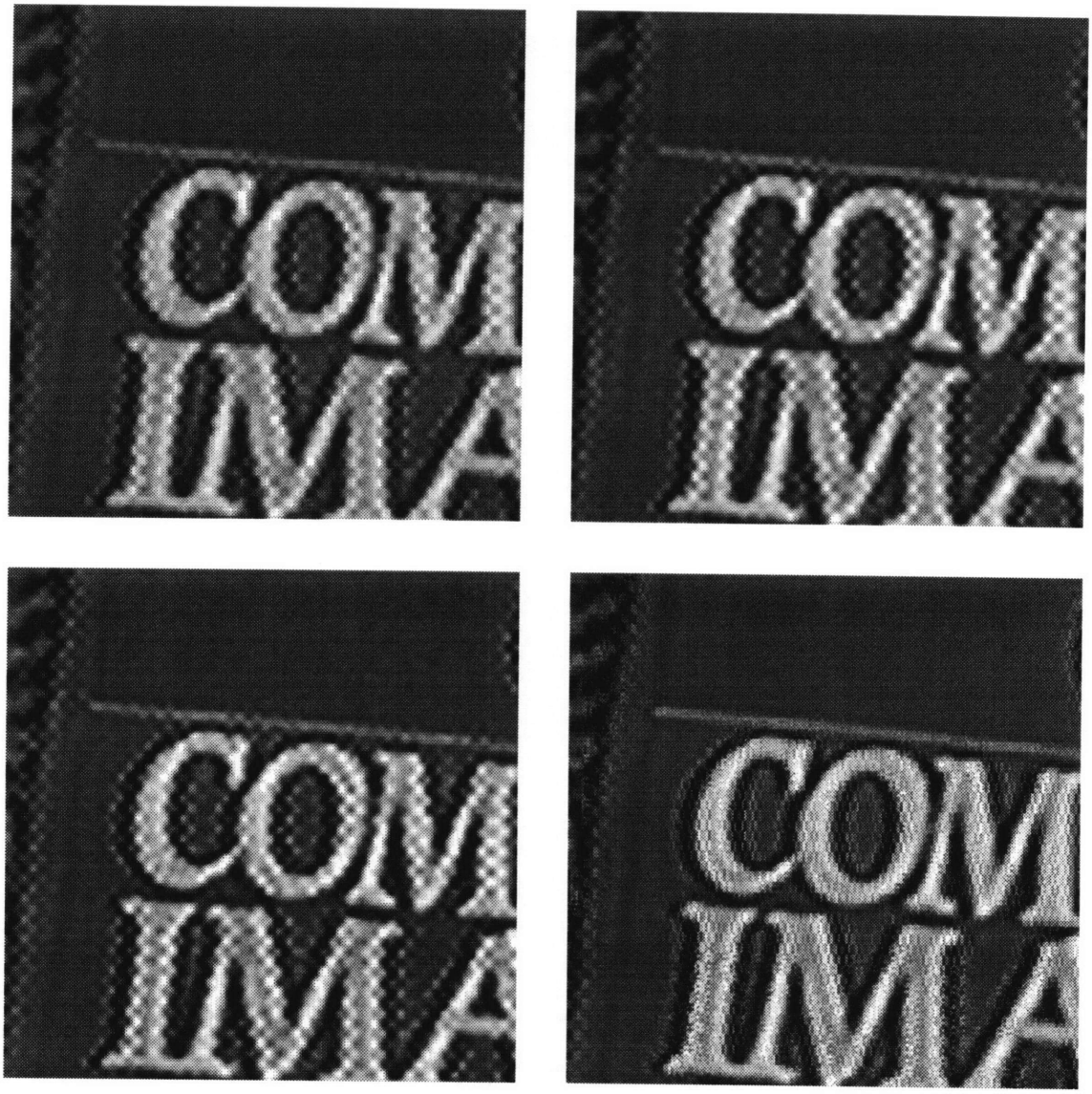

Figure 4-15: (a) Bilinear (top left), (b) Bicubic (top right), (c) Piecewise cubic (bottom left), (d) Multi-frame resolution enhancement (bottom right). The advantage of the multi-frame technique is that you are able to produce sharper edges while interpolation techniques tend to blur the results and smooth the image.

Figure 4-16: Original image frame

level pyramid with 6 GNC stages is used to do motion estimation. Figure 4-20 shows the results of multi-frame resolution enhancement. There are noticeable artifacts around object boundaries in the image as a result of incorrect motion estimates at some pixels. The underlying assumption of accurate motion estimation for every pixel is not met. To reduce some of the artifacts, a 3 point separable median filter is applied. Median filtering is used to reduce noise since it will not smooth image edges. The resulting image is shown in Figure 4-21.

Figure 4-17: Bilinear interpolation

Figure 4-18: Bicubic interpolation

Figure 4-19: Piecewise cubic interpolation

Figure 4-20: Multi-frame resolution enhancement

Figure 4-21: Multi-frame resolution enhancement and median filter

Chapter 5

Conclusion

This thesis developed a method for video resolution enhancement when multiple motions are present in the image sequence. The motion estimation used is based on the algorithm presented in [3]. The algorithm is modified by the use of a Laplacian pyramid structure and applying all stages of the GNC algorithm for global minimization at each pyramid level before moving to the next level. In addition, the original method was not used in the context of resolution enhancement. Median filtering is added to reduce degradation from incorrect motion estimates.

The results from multi-frame resolution enhancement are compared with several interpolation techniques. Interpolation methods which have low computational expense smooth edges in the image and are not able to truly increase resolution in the image.

Using multiple image frames to do resolution enhancement has the potential to produce a sharper image. Actual scene information is used, and edges in the image can be preserved. The premise for multi-frame resolution enhancement is based on accurate pixel displacement estimation for every pixel in the image. Consequently, with current motion estimation algorithms, resolution enhancement may result in some image artifacts, and additional image restoration techniques must be used when necessary.

Bibliography

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual moiton. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimation two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896, September 1992.

[3] M. J. Black. *Robust Incremental Optical Flow.* PhD thesis, Yale University, New Haven, CT, 1992. Research Report YALEU/DCS/RR-923.

[4] M.J. Black and P. Anandan. Constraints for the early detection of discontinuity from motion. In *Proc. AAAI90*, pages 1060–1066, Boston, MA, 1990.

[5] A. Blake and A. Zisserman. *Visual Reconstruction.* The MIT Press, Cambridge, Massachusetts, 1987.

[6] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, April 1983.

[7] P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, New Jersey, October 1991.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[9] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[10] M. Irani and S. Peleg. Improving resolution by image registration. In *CVGIP: Graphical Models and Image Processing*, volume 53, pages 231–239, May 1991.

[11] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, May 1992.

[12] A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[13] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, Ann Arbor, MI, June 1988.

[14] S.P. Kim, N.K. Bose, and H.M. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):1013–1027, June 1990.

[15] S.P. Kim and Wen-Yu-Su. Recursive high resolution reconstruction of blurred multiframe images. *IEEE Transactions on Image Processing*, 2(4):534–539, October 1993.

[16] J.S. Lim. *Two-Dimensional Signal and Image Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[17] S. Peleg and H. Rom. Motion based segmentation. In *International Conference on Pattern Recognition*, pages 109–113, Atlantic City, June 1990.

[18] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. In *CVGIP: Graphical Models and Image Processing*, volume 54, pages 181–186, March 1992.

[19] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Los Alamitos, California, 1990.